STRATHMORE INSTITUTE OF MATHEMATICAL SCIENCES
Master of Science in Mathematical Finance and Risk Analytics
Assignment
MFI 8302: Computational Methods in Finance

Bernice W Kangethe

August 2024

1 Variance Reduction

1.1 Let X be an arbitrary random variable.Let f be a monotonically increasing function and g a monotonically decreasing function. Show that f (X) and g(X) are negatively correlated. That is, Cov [f(X)g(X)]$\leq$ 0 Hint: Let Y be an independent random variable that has the same distribution as X.Argue that [f (X)f(Y)] , [g(X) g(Y)] 0 and then take expected value on both sides.

SOLUTION

- Consider the expression [f (X)-f(Y)][g(X) g(Y)] and expand it.This gives f(X)g(X) - f(X)g(Y) - f(Y)g(X) - f(Y)g(Y). Considering random Variable Y with the same distribution as X and expanding the expression [f (X)-f(Y)][g(X) g(Y)]
- Take the expected value of the expanded expression
E[[f (X)-f(Y)][g(X) g(Y)]] = E[f(X)g(X)]-E[f(X)g(Y)]-E[f(Y)g(X)]+E[f(Y)g(Y)]]
- Since X and Y are identically distributed and independent,
(Given that Y is an independent random variable with the same distribution as X, the cross terms E[f(X)g(Y)] and E[f(Y)g(X)] can be simplified to the product of their expectations.)

$$E[f(X)g(Y)] = E[f(X)]E[g(Y)] = E[f(X)]E[g(X)]$$
$$E[f(Y)g(X)] = E[f(Y)]E[g(X)] = E[f(X)]E[g(X)]$$

- Thus,the expression simplifies to
E[f(X)g(X)] - E[f(X)]E[g(X)] - E[f(X)]E[g(X)] + E[f(X)g(X)] = 2E[f(X)g(X)] - 2E[f(X)][g(X)] , which is Non-positive since f(X) and g(X) are monotonically increasing and decreasing, respectively.
This shows that the covariance is non-positive, proving that 2E[f(X)g(X)] - 2E[f(X)][g(X)]$\leq$ and f(X) and g(X) are negatively correlated.
- Hence cov [f(X)g(X)]$\leq$ 0

1.2 Let X and Y be two arbitrary random variables. Define the conditional variance of X given Y by Var[XY] = E[X^{2}Y] (E[XY])^2 Show that Var[XY] 0 and the variance decomposition formula Var[X] = E[Var[XY]]+Var[E[XY]]

SOLUTION
Var[X$|Y] = E[X - E([X—$Y])^2|Y] The conditional variance of X given Y is defined as the expected value of the squared difference between X and its conditional expectation given Y.
E[X^2|Y]-(E[X|])$^2 \geq 0$. By definition, variance is always non-negative. Therefore, the Var[X|Y] conditional variance is non-negative, which implies that E[X^2|Y]-(E[X|])$^2 \geq 0$
This is the variance decomposition formula states that the total variance of X can be decomposed into the expected value of the conditional variance of X given Y and the variance of the conditional expectation of X given Y. Var(X)=E[Var[X|Y]+ Var(E[X|Y])
Substitution of conditional variance into the variance decomposition formula.
Var(X)=E[E[X^2|Y]-(E[X| Y])^2] + Var(E[X|Y]) .
Using the law of total variance, the variance of X can be decomposed into the expected value of the conditional variance of X given Y plus the variance of the expected value of X given Y, written as
Var(X)=E[E[X^2|Y]-(E[X| Y])^2] + Var(E[X|Y])

1.3 The variance decomposition formula in Question 2 above suggests the following variance reduction technique. Consider the problem of estimating = E[h(X)]. Let Y be a random variable for which f(Y) = E[h(X)Y] is explicitly known. Let Y1...Yn be iid copies of Y, and define the estimate to be

$$\mu = \frac{1}{n}\sum_{i}^{n} f(Y_i)$$

Show that μ is unbiased and its variance is always no greater than the variance of the plain Monte Carlo estimate with the same sample size. This variance reduction technique is called the method of conditioning. Discuss the difference between the method of conditioning and stratified sampling.

E[θ] = E[$\sum_{i=1}^{n} f(Y_i)$] = $1/n*\sum_{i=1}^{n} E[f(Y_i)]$ =θ
To show that the estimate θ is unbiased, you need to take the expectation of . Since the Yi are iid, the expectation of the sum is the sum of the expectations.
Var(θ) = Var(1/n * $\sum_{i=1}^{n} f(Y_i)) = 1/n^2 * \sum_{i=1}^{n} Var(f(Y_i)) = 1/n * Var(f(Y))$.

To find the variance of θ, use the fact that the variance of a sum of iid random variables is the sum of their variances. Since you are averaging n terms, you divide by n^2.
Var($\theta_M C) = Var(1/n * \sum_{i=1}^{n} h(X_i)) = 1/n * Var(h(X))$
The variance of the plain Monte Carlo estimate $\theta_M Cisgivenbythevarianceoftheaverageofniidcopiesofh(X)$.
Var(θ) = 1/n * Var(f(Y)) 1/n * Var(h(X)) = Var($\theta_(MC)$)
Since f(Y) = E[h(X)—Y], the variance of f(Y) is less than or equal to the variance of h(X) by the law of total variance. Thus, the variance of θ is less than or equal to the variance of the plain Monte Carlo estimate.
The method of conditioning involves using a known conditional expectation to reduce variance, while stratified sampling involves dividing the population into strata and sampling from each stratum to ensure representation.

1.3.1 Discuss the difference between the method of conditioning and stratified sampling. The method of conditioning uses conditional expectations to achieve variance reduction, whereas stratified sampling ensures that all parts of the population are represented in the sample.

SOLUTION
The estimate θ is unbiased, and its variance is always no greater than the variance of the plain Monte Carlo estimate. This variance reduction technique is called the method of conditioning. The method of conditioning

involves using a known conditional expectation to reduce variance, while stratified sampling involves dividing the population into strata and sampling from each stratum to ensure representation.

1.4 Suppose that we wish to estimate the price of a spread call option with maturity T and payoff H = (X_T -Y_T - K)+. Using the control variate method, write a function to estimate the option price by the method of conditioning where the function should have input parameters X_0 = 50, Y_0 = 45, r = 0.05,σ_2=0.2 ,σ_2=0.3 ρ=05 ,T =1 with n=10000 and K =0.5, 1.0, respectively. Report your estimate and standard error for Monte Carlo simulations.

SOLUTION

The underlying asset prices X_t and Y_t are modeled using Geometric Brownian Motion (GBM) defined by.
dXt = μ Xt dt + σ Xt dW1t,
dYt = μ Yt dt + σ Yt dW2t where μ is the drift, σ is the volatility, and dW1t and dW2t is the increment of a Wiener processes with correlation rho.

Calculate the payoff H for each simulated path at maturity T.For each simulated path, calculate the payoff of the spread call option at maturity T. The payoff function H = max,0) For each simulated path, calculate the payoff of the spread call option at maturity T. The payoff function H is given by (X_T - YT - K), where a+ denotes the positive part of a.

Using the GBM models, simulate a large number of paths for the underlying asset prices X_t and Y_t over the time from t to T. Each path represents a possible realization of the asset price movements and is generated by discretizing the time interval and applying the GBM formulas to calculate the prices at each time step.

Compute the discounted expected payoff under the risk-neutral measure Q using the control variate method. To find the option's present value, discount the expected payoff back to time t using the risk-free rate r. The Monte Carlo estimate of the expectation is the average of the discounted payoffs across all simulated paths. The control variate method helps reduce the variance of the estimate.

The Monte Carlo estimate of the option price is obtained by averaging the discounted payoffs from all simulated paths, using the control variate method to reduce variance.

1.4.1 Report of Option Price estimate and standard error for Monte Carlo simulation.

1.5 Discuss the main steps used in Monte Carlo methods to approximate the expectation V(S0 0) = $e^r(T-t)E_Q$ [$\Psi(S_T)$—S_0 = s] where Ψ(ST) is the payoff an average price Asian option at maturity T and St is the price of the underlying process that follows a Geometric Brownian motion.

Solution

- In Monte Carlo simulations for option pricing, the first step is to define the stochastic process that the underlying asset follows. For an average price Asian option, the underlying asset price St is typically modeled using a Geometric Brownian Motion (GBM) defined by.dSt = μ St dt + σ St dWt, where μ is the drift, σ is the volatility, and dWt is the increment of a Wiener process.
- Using the GBM model, simulate a large number of paths for the underlying asset price over the time from t to T. Each path represents a possible realization of the asset price movement and is generated by discretizing the time interval and applying the GBM formula to calculate the price at each time step.
- For each simulated path, calculate the payoff of the Asian option at maturity T. The payoff function ψ(ST) depends on the specific type of Asian option. For an average price option, the payoff is typically a function of the average of the underlying asset prices along the path.
- To find the option's present value, discount the expected payoff back to time t using the risk-free rate r. The expectation is taken under the risk-neutral measure Q, which means that the drift μ in the GBM is replaced by the risk-free rate r. The Monte Carlo estimate of the expectation is the average of the discounted payoffs across all simulated paths.The Monte Carlo approximation of the expectation V(S_0, 0) is the average of the discounted payoffs from all the simulated paths of the underlying asset price.

+ Code + Text All changes saved

```python
import numpy as np

def estimate_spread_call_option_price(X0, Y0, r, sigma1, sigma2, rho, T, n, K):
    # Set random seed for reproducibility
    np.random.seed(0)

    # Generate correlated random numbers
    Z1 = np.random.normal(0, 1, n)
    Z2 = rho * Z1 + np.sqrt(1 - rho**2) * np.random.normal(0, 1, n)

    # Simulate terminal values of X_T and Y_T
    X_T = X0 * np.exp((r - 0.5 * sigma1**2) * T + sigma1 * np.sqrt(T) * Z1)
    Y_T = Y0 * np.exp((r - 0.5 * sigma2**2) * T + sigma2 * np.sqrt(T) * Z2)

    # Calculate the payoff for each simulation
    payoffs = np.maximum(X_T - Y_T - K, 0)

    # Control variate - known expectation (risk-neutral expectation of (X_T - Y_T))
    # E[X_T - Y_T] = X0 * exp(r * T) - Y0 * exp(r * T)
    control_variate = X_T - Y_T - (X0 * np.exp(r * T) - Y0 * np.exp(r * T))

    # Calculate the control variate estimator
    alpha = -np.cov(payoffs, control_variate)[0, 1] / np.var(control_variate)
    control_variate_adjusted_payoffs = payoffs + alpha * control_variate

    # Discount the adjusted payoffs to present value
    option_price = np.mean(control_variate_adjusted_payoffs) * np.exp(-r * T)

    # Calculate standard error
    standard_error = np.std(control_variate_adjusted_payoffs) / np.sqrt(n)

    return option_price, standard_error
```

```python
# Input parameters
X0 = 50
Y0 = 45
r = 0.005
sigma1 = 0.2
sigma2 = 0.3
rho = 0.5
T = 1
n = 10000
K = 0.5

# Estimate the option price and standard error
option_price, standard_error = estimate_spread_call_option_price(X0, Y0, r, sigma1, sigma2, rho, T, n, K)

option_price, standard_error
```

```
(7.506685702718078, 0.039263311415054306)
```